1980

Error Detecting and Correcting Codes

Hamming, Richard W.

Monterey, California: Naval Postgraduate School

ERROR DETECTING AND CORRECTING CODES

Although the telegraph was the first system to use codes for transmitting information electrically, and is still their most extensive user, other applications have assumed greater and greater importance in recent years. All common-control telephone switching systems use code for transmitting information between circuits; and electronic digital computers, which are daily coming into more extensive use, employ them exclusively, not only for transmitting information between component circuits, but in carrying out the actual computations. Moreover, some attention has always been paid to methods of detecting errors, caused perhaps by faulty contacts, by open circuits, or by disturbances induced by outside sources. The wordcount at the end of a telegram is one simple method, and the two-out-of-five method employed widely in telephone switching is another and very effective way of making it impossible for the more common types of error to escape detection. The next step in these efforts to eliminate errors, is to devise methods of not only detecting but of correcting any preassigned number of errors that may occur.

In relay computing machines, of which a number have been designed and built by the Laboratories*, a binary form of code is generally employed. A symbol in such a code may be represented by a succession of 1's or 0's - the 1's representing the transmission of current and the 0's no current. A binary code is particularly suited for use with relay circuits since relays may be in either of

- -

*RECORD, <u>October</u>, 1940, <u>page</u> V; <u>December</u>, 1946, <u>page</u> 457; <u>January</u>, 1947, <u>page</u> 5; <u>February</u>, 1947, <u>page</u> 49; <u>May</u>, 1948, <u>page</u> 208.

two conditions; operated or non-operated. The former corresponds to a 1 in the code, and the latter to a 0. It is also well suited to represent electronic circuits involving flip-flops and to systems employing perforated tapes.

The succession of the digits 0 or 1 in a binary symbol represents a number just as the succession of the digits 0 to 9 represents a number in the decimal system. In the latter system, the digits from right to left are interpreted as the coefficients of successive powers of 10, while in a binary number they are the coefficients of the successive powers of 2. The decimal number 237, for example, is interpreted as $2 \times 10^2 + 3 \times 10^1 + 7 \times 10^0$, while the binary number 1101 is interpreted as $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$.

In transmitting and operating with symbols in a binary code one may send the individual 1's and 0's one after the other in sequence, or all at the same time in parallel. The Laboratories relay machines operated, in the main, in parallel. As far as the codes discussed here are concerned there is no reason to distinguish the two types since the methods and arguments used apply equally well to both.

With binary codes, it is very easy to detect errors by means of what is known as a parity check. While it is not essential, the assumption that each code symbol is always made to include the same number of binary digits is both of great convenience in exposition and represents the situation in almost all practical situations thus far contemplated. A code consisting of symbols of this type is known as a systematic code. If a six-digit symbol were to be employed, a symbol for the number 1 would be 000001, while the symbol for the

number 53 would be 110101. To provide a parity check for such a code, the number of digits would be increased to seven; the first six digits would carry the information, and the final place would be for the parity check. Into this last place at the sending end of the circuit would be placed a 0 or a 1 to make the total number of 1's either even or odd. If the parity check digit is used to make the total number of 1's even, it is called an even parity check; for the sake of simplicity this will be the only type considered here. With an even parity check, the symbol transmitted for number 53 would be 1101010 - a 0 being put in the last position since there are four, and thus an even number of, 1's in the information positions.

Suppose now that due to some disturbance in the circuit, this symbol were received as 1001010 - the 1 in the second position having been lost in transmission. At the receiving end, a parity check on this symbol would reveal that there should be a 1 in the last, or check, position, while a 0 is actually found there. It is evident, therefore, that the symbol has been mutilated some way in transmission. The receiving circuit would be designed to indicate the fact by giving an alarm, or ceasing to operate, or both.

Such a scheme would prevent a computer from giving erroneous results, but if it were unattended at the time the trouble arose, it would remain idle until the operating force returned and located and corrected the trouble. If it were possible, however, for the error detecting circuit not only to detect an error but to discover its exact position, it could be corrected by replacing the digit in that position by a 1 if it were a 0 or by a 0 if it were a 1. After the correction

had been made, action could be continued as though no error had occurred, and no operating time of the machine would be lost. This is what the present studies have made possible. We will show how to use parity checks to identify precisely the position of an error and thus allow it to be corrected. In principle a code may be devised to correct any preassigned number or errors; if fact the correction of single errors in a code symbol seems to be as far as it is economically sound to go at present. Thus we shall discuss only single error correcting codes.

There is no reason, of course, why more than one parity check cannot be used, each being applied over some of the digits of the symbol, and the method takes advantage of this fact in determining the exact position of the error. The proposal is to apply a number of parity checks in a specified order at the sending end, and to put in the place reserved for each parity check a 0 when a check shows an even number of 1's in the checked positions, and a 1 when it shows an odd number. At the receiving end, the parity checks would again be applied in the same order this time applied over the same positions <u>plus</u> the check position, and each time these parity checks show an even number of 1's, a 0 will be recorded, while each time there checks show an odd number of 1's, a 1 will be recorded. These latter digits if written from right to left may be viewed as a binary number and are known as the checking number. The number of checks and the positions in which the results of each check are placed are so selected that this checking number is the number of the position in the original symbol in which the error has occurred.

In order to fix ideas let k be number of parity checks to be used - the value of k will be determined a little later. Just as a decimal number of k digits can indicate any of 10^k values, so a binary number of k digits can indicate any of 2^k values. It is thus necessary to make k large enough so that 2^k will be great enough to indicate any of the positions of the symbol plus one value to indicate no error. Now if n is allowed to represent the total number of digits in the symbol, and k the number of positions for the parity checks, then there will remain $n - k = m$ positions for the information. Since k is to be large enough to indicate any position of the symbol plus one indication for no error, 2^k must be equal to or greater than $(n + 1)$. Since, $k = n - m$, this relationship may be written as $2^{(n - m)} \geq (n + 1)$, or, since $2^{(n - m)} = \frac{2^n}{2^m}$, this readily converts to $\frac{2^n}{n + 1} \geq 2^m$.

Whenever the number of digits in a symbol is increased to obtain parity checks, however, the efficiency of transmission is reduced, since more digits must be transmitted than are needed to convey the required information. This decrease in efficiency may be measured by the ratio, R, of the total number of digits, n, to the number, m, required to transmit the information. This ratio, $R = u/m$, is called the redundancy. In calculating the corresponding values of k, m, and n from the above expression, therefore, only the values of n and k are used that give the lowest redundancy for each value of m. The values of k, m, and n that meet this requirement are given in Table I.

Having thus determined the number of check positions that will be required for any number of information positions, it is necessary